Lec 4 Graph-based Register Allocation

→ Interference graph is not a fixed notion
it depends on how we approximate the interference

Reg alloc steps

1. Construct interference graph
2. Find k-colouring, with k as small as possible
3. Assign ~13 colours, and respective temps, to registers
4. Spill remaining temps to stack

Graph Colouring

Problem Definition (decision version)

Given graph $G = (V, E)$ and k colours, decide if there is a colouring of all $v \in V$ s.t. $\forall \{u, v\} \in E$, u, v have different colour

This is NP-complete (for $k \geq 3$)

But solving reg alloc doesn't require solving graph colouring, we may be able to modify the programme or construct the graph some certain ways.

Real problem (reg alloc as decision problem):

For Turing-complete language, it's undecidable wether a prog. p has an equivalent prog. p' s.t. p' uses k registers and no temp

Greedy Colouring

→ Turns out most interference graphs are chordal

Assume / Define :
- Colours $= C = \{1, 2, 3, \dots, k\}$
- $N(v)$ is nbors of $v \in V$
- V is ordered $(v_1, \dots, v_n)$

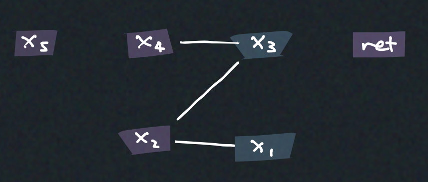
Problem : Input : $G = (V, E)$, $V = (v_1, \dots, v_n)$
Output : $\Delta : V \to C$

Greedy colouring alg

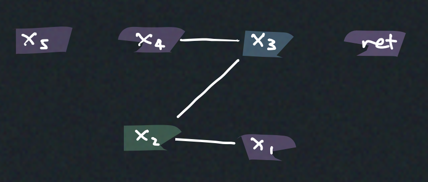
for v_i in $v_1, \dots, v_n$:
  let $c \in C$ be lowest colour s.t. $\nexists u \in N(v_i)$, $\Delta(u) = c$
  $\Delta(v_i) := c$

Ex.

$C = \{ \quad \}$



← $V = (x_5, x_4, x_3, x_2, x_1, ret)$
↑
good order



← $V = (x_1, x_4, x_3, x_2, x_5, ret)$
↑
bad order

→ So we want to find a good ordering

Thm There exists an ordering that produces an optimal colouring that produces an ordering
└ proof is that one can start with optimal Δ^* and sort V by $\lambda\, v \Rightarrow \Delta^*(v)$ (or something similar)

→ Then question becomes finding optimal colouring

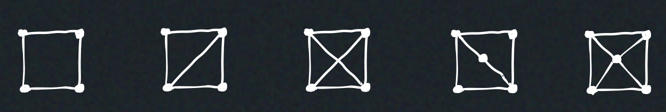
Chordal Graphs

Def G is chordal if $\forall$ cycles c in G, length of $c \geq 4 \Rightarrow \exists$ a chord $e \in E$ btwn two verts in c s.t. e is not in c.

Def A chord connects two verts in a cycle without being part of the cycle

Exs

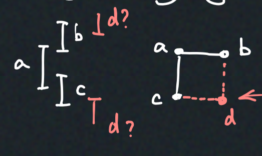
G



Chordal? ✗ ✓ ✓ ✗ ✗

Study
- 95% of all interference graphs are chordal
- in SSA, all interference graphs are chordal
  └ static single assignment

Intuition - to get long cycle without chord



← how do you get this ?
not possible in SSA
unusual in normal programme

Maximum Cardinality Search (MCS)

Input : $G = (V, E)$
Output : ordering $v_1, \dots, v_n$

MCS Algorithm ─ optimal if input is chordal
else good estimation
└ $O(m+n)$ if good data structure

for $v \in V$, set $w(v) \leftarrow 0$
└ weight
$W \leftarrow V$
for i in $1, \dots, n$:
  pick $v \in W$ with max weight
  $v_i \leftarrow v$
  for $u \in N(v) \cap W$:
    $w(u) \leftarrow w(u) + 1$
  $W \leftarrow W \setminus \{u\}$

Thm MCS returns a simplicial elimination ordering if G is chordal

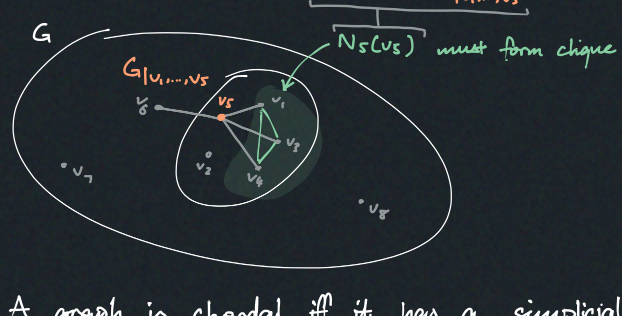
Def v is <u>simplicial</u> in G if $N(v)$ is a clique

Def A <u>simplicial elim. ordering</u> is ordering $v_1, \dots, v_n$ s.t. $\forall v_i$, v_i is simplicial in $G|_{v_1, \dots, v_i}$, the subgraph induced by

Ex.

$v_1, \dots, v_i$

$\underbrace{N(v_5) \text{ in } G|_{v_1, \dots, v_5}}_{N_5(v_5)}$ must form clique



Thm A graph is chordal iff it has a simplicial elim. ordering

Proof

(⇒) hard
(⇐) homework — induct

Sidenote the graph is perfect (... whatever that means)

Thm The greedy colouring algorithm finds optimal colouring if we run it with simplicial elim. ordering

Proof Let k be num colours used
Observe :
1. $k \leq \max_i |N_i(v_i)| + 1$ ← in worse case we pick new colour
   ↑ nborhood in i^{th} induced subgraph
   worse case all nbors use different colour
2. min # colours $\geq \max_i \underline{|N_i(v_i)|} + 1$ ⊂ plus $\Delta(v_i)$'s colour
   ↑ because this clique must have unique colours

Thm MCS returns simplicial elim. ordering iff G is chordal

Proof hard

Summary

1. Build interference graph
2. Order vertices with MCS
3. Colour graph with greedy alg
4. Spill if too many colours

Spilling

Strategies
1. Least used temp should spill
2. Loop nesting : prioritise inner loop
3. Just spill highest colours — since greedy tries to use lowest colour